Assignment 3

Student Name:

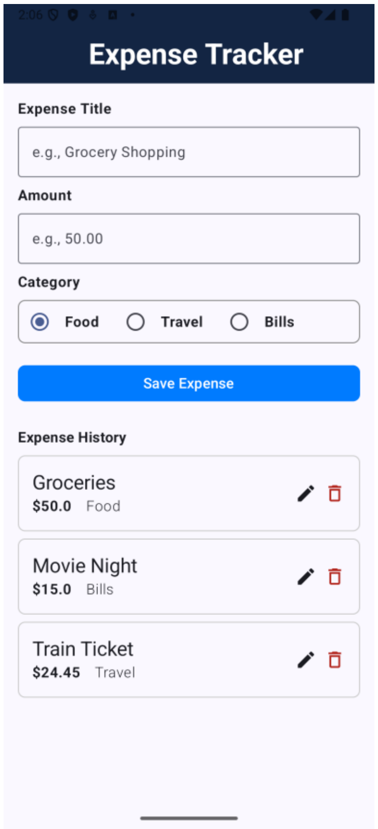
Parth Gajjar (8959999)

Course:

PROG3320 - Systems Project Planning & Initiation

Faculty: Thomas Methew

Screenshots (Light and Dark Color Scheme):



2:06

Expense Tracker

Expense Title

e.g., Grocery Shopping

Amount

e.g., 50.00

Category

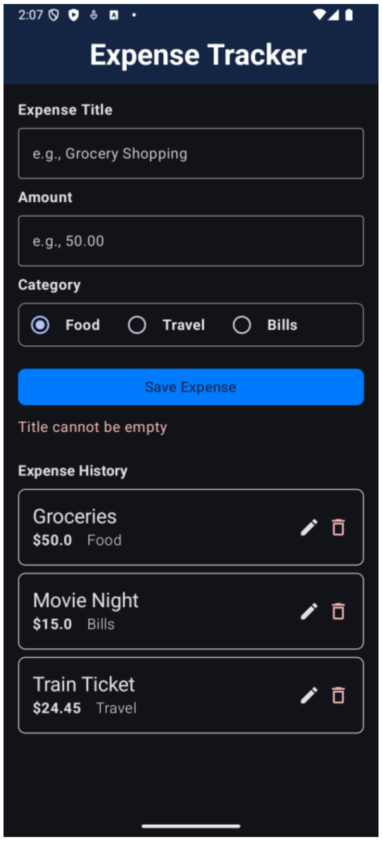
● Food ○ Travel ○ Bills

Expense History

Groceries
$50.0 Food

Movie Night
$15.0 Bills

Train Ticket
$24.45 Travel



2:07

Expense Tracker

Expense Title

e.g., Grocery Shopping

Amount

e.g., 50.00

Category

● Food ○ Travel ○ Bills

Title cannot be empty

Expense History

Groceries
$50.0 Food

Movie Night
$15.0 Bills

Train Ticket
$24.45 Travel

Code:

Expense.kt:

```
package com.example.expensetracker8959999.data

// Enum representing expense categories
enum class ExpenseCategory {
  Food,
  Travel,
  Bills
}

// Class representing an expense
data class Expense(
  val id: Int = 0,
  val title: String,
  val amount: Double,
  val category: ExpenseCategory
)
```

ExpenseDatabaseHelper.kt:

```
package com.example.expensetracker8959999.data

import android.content.ContentValues
import android.content.Context
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper

class ExpenseDatabaseHelper(context: Context):
SQLiteOpenHelper(context, "expenses", null, 1) {
  override fun onCreate(db: SQLiteDatabase?) {
    db?.execSQL("""
      CREATE TABLE expenses(
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        title TEXT,
        amount REAL,
        category TEXT
      )
    """.trimIndent())
  }
```

```
override fun onUpgrade(db: SQLiteDatabase?, p1: Int, p2: Int) {
    db?.execSQL("DROP TABLE IF EXISTS expenses")
    onCreate(db)
}

fun insertExpense(title: String, amount: Double, category: ExpenseCategory) {
    val db = writableDatabase
    val values = ContentValues().apply {
        put("title", title)
        put("amount", amount)
        put("category", category.toString())
    }
    db.insert("expenses", null, values)
    db.close()
}

fun getAllExpenses(): List<Expense> {
    val expenses = mutableListOf<Expense>()
    val db = readableDatabase
    val cursor = db.rawQuery("SELECT * FROM expenses", null)
    while (cursor.moveToNext()) {
        expenses.add(
            Expense(
                cursor.getInt(0),
                cursor.getString(1),
                cursor.getString(2).toDouble(),
                ExpenseCategory.valueOf(cursor.getString(3))
            )
        )
    }
    cursor.close()
    db.close()
    return expenses
}

fun updateExpense(id: Int, title: String, amount: Double,  category: ExpenseCategory) {
    val db = writableDatabase
    val values = ContentValues().apply {
        put("title", title)
        put("amount", amount)
        put("category", category.toString())
    }
```

```
        db.update("expenses", values, "id=?", arrayOf(id.toString()))
        db.close()
    }
    fun seedData() {
        insertExpense("Groceries", 50.0, ExpenseCategory.Food)
        insertExpense("Bus Ticket", 2.5, ExpenseCategory.Travel)
        insertExpense("Movie Night", 15.0, ExpenseCategory.Bills)
    }

    fun deleteExpense(id: Int){
        val db = writableDatabase
        db.delete("expenses", "id=?", arrayOf(id.toString()))
        db.close()
    }
}
```

ExpenseCard.kt:

```
package com.example.expensetracker8959999.components

import androidx.compose.foundation.border
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Row
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.material.icons.Icons
import androidx.compose.material.icons.filled.Edit
import androidx.compose.material.icons.outlined.Delete
import androidx.compose.material3.Icon
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import com.example.expensetracker8959999.data.Expense
import com.example.expensetracker8959999.data.ExpenseCategory
import com.example.expensetracker8959999.ui.theme.ExpenseTracker8959999Theme


// Expense Card Composable for displaying expense details

@Composable
fun ExpenseCard(expense: Expense, onEditClick: (id: Int) -> Unit, onDeleteClick: (id: Int) -> Unit) {
    Row(
        modifier = Modifier
            .border(
                width = 1.dp,
                color = Color.LightGray,
                shape = RoundedCornerShape(8.dp)
            )
            .fillMaxWidth()
            .padding(16.dp),
        horizontalArrangement = Arrangement.spacedBy(8.dp),
        verticalAlignment = Alignment.CenterVertically,

    ) {
        Column(
            modifier = Modifier.weight(1f)
        ) {
            Text(
                text = expense.title,
                style = MaterialTheme.typography.titleLarge
            )
            Row(
                horizontalArrangement = Arrangement.spacedBy(16.dp)
            ) {
                Text(
                    text = "$${expense.amount}",
                    fontWeight = FontWeight.Bold
                )
                Text(
                    text = expense.category.toString(),
                    fontWeight = FontWeight.Light
                )
            }
        }

        Icon(
            imageVector = Icons.Filled.Edit,
            contentDescription = "Edit",
            modifier = Modifier
                .clickable(
                    onClick = { onEditClick(expense.id) }
                )
        )

        Icon(
            imageVector = Icons.Outlined.Delete,
            contentDescription = "Delete",
            modifier = Modifier
                .clickable(
                    onClick = { onDeleteClick(expense.id) }
                ),
            tint = MaterialTheme.colorScheme.error
        )

    }
}

@Preview(showBackground = true)
@Composable
fun ExpenseCardPreview() {
    val expense = Expense(
        id = 1,
        title = "Groceries",
        amount = 50.0,
        category = ExpenseCategory.Food
    )
    ExpenseTracker8959999Theme {
        ExpenseCard(expense = expense, onEditClick = {}, onDeleteClick = {})
    }
}
```

ExpenseList.kt:

```
package com.example.expensetracker8959999.components

import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.lazy.items
import androidx.compose.runtime.Composable
import androidx.compose.ui.unit.dp
import com.example.expensetracker8959999.data.Expense


// Expense List Composable to display a list of expenses
@Composable
fun ExpenseList(
    expenses: List<Expense>,
    onEdit: (Expense) -> Unit,
    onDelete: (Expense) -> Unit
) {
    LazyColumn(
        verticalArrangement = Arrangement.spacedBy(8.dp)
    ) {
        items(expenses, key = { it.id }) { expense ->
            ExpenseCard(
                expense = expense,
                onEditClick = { onEdit(expense) },      // Get from ExpenseCard
                onDeleteClick = { onDelete(expense) }   // Get from ExpenseCard
            )
        }
    }
}
```

ExpenseForm.kt:

```
package com.example.expensetracker8959999.components

import androidx.compose.foundation.BorderStroke
import androidx.compose.foundation.border
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Row
import androidx.compose.foundation.layout.Spacer
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.height
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.foundation.text.KeyboardOptions
import androidx.compose.material3.Button
import androidx.compose.material3.ButtonDefaults
import androidx.compose.material3.OutlinedTextField
import androidx.compose.material3.RadioButton
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.input.KeyboardType
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.expensetracker8959999.R
import com.example.expensetracker8959999.data.ExpenseCategory
import com.example.expensetracker8959999.features.expensetracker.ExpenseTrackerViewModel

// Expense Form Composable for Adding/Editing Expense
@Composable
fun ExpenseForm(viewModel: ExpenseTrackerViewModel) {
  Column(
    modifier = Modifier.fillMaxWidth(),
    verticalArrangement = Arrangement.spacedBy(8.dp)
  ) {
    // Title Input
    Text(stringResource(R.string.txtField_title), fontWeight = FontWeight.Bold)
    OutlinedTextField(
      value = viewModel.expenseTitle,
      onValueChange = { viewModel.updateExpenseTitle(it) },
      modifier = Modifier.fillMaxWidth(),
      singleLine = true,
      placeholder = { Text("e.g., Grocery Shopping") }
    )

    // Amount Input
    Text(stringResource(R.string.txtField_amount), fontWeight = FontWeight.Bold)
    OutlinedTextField(
      value = viewModel.expenseAmount,
      onValueChange = { viewModel.updateExpenseAmount(it) },
      modifier = Modifier.fillMaxWidth(),
      keyboardOptions = KeyboardOptions(keyboardType = KeyboardType.Number),
    singleLine = true,
    placeholder = { Text("e.g., 50.00") }
)

// Category Selection
Text(stringResource(R.string.txtField_category), fontWeight = FontWeight.Bold)
val categories = ExpenseCategory.entries
Row(
    modifier = Modifier
        .fillMaxWidth()
        .border(
            BorderStroke(1.dp, Color.Gray),
            shape = RoundedCornerShape(8.dp)
        ),
    verticalAlignment = Alignment.CenterVertically
) {
    categories.forEach { category ->
        Row(
            verticalAlignment = Alignment.CenterVertically,
            modifier = Modifier
                .clickable { viewModel.updateSelectedCategory(category) }
                .padding(end = 16.dp)
        ) {
            RadioButton(
                selected = (category == viewModel.selectedExpenseCategory),
                onClick = { viewModel.updateSelectedCategory(category) }
            )
            Text(
                text = category.name, // Using enum name (e.g., Food, Travel)
                modifier = Modifier.padding(start = 4.dp), fontWeight = FontWeight.Bold
            )
        }
    }
}

Spacer(modifier = Modifier.height(4.dp))

// Save Button
Button(
    onClick = { viewModel.saveExpense() },     // Get from viewModel
    modifier = Modifier.fillMaxWidth(),
    shape = RoundedCornerShape(8.dp),
            colors = ButtonDefaults.buttonColors(
                containerColor = Color(0xFF007BFF) // Blue
            )
        ) {
            Text(stringResource(R.string.save_button), fontSize = 16.sp)
        }
    }
}
```

ExpenseTrackerViewModel:

```
package com.example.expensetracker8959999.features.expensetracker

import android.app.Application
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.setValue
import androidx.lifecycle.AndroidViewModel
import com.example.expensetracker8959999.data.Expense
import com.example.expensetracker8959999.data.ExpenseCategory
import com.example.expensetracker8959999.data.ExpenseDatabaseHelper


class ExpenseTrackerViewModel(application: Application): AndroidViewModel(application) {

    private val dbHelper = ExpenseDatabaseHelper(application)

    var expenses by mutableStateOf<List<Expense>>(dbHelper.getAllExpenses())
        private set
    var expenseTitle by mutableStateOf("")
        private set
    var expenseAmount by mutableStateOf("")
        private set

    var selectedExpenseCategory by mutableStateOf(ExpenseCategory.Food)
        private set

    var showError by mutableStateOf(false)
        private set
    var errorMessage by mutableStateOf("")
        private set
```

```
var editingId: Int? by mutableStateOf(null)
    private set

fun updateExpenseTitle(newTitle: String) {
    expenseTitle = newTitle
}

fun updateExpenseAmount(newAmount: String) {
    expenseAmount = newAmount
}

fun updateSelectedCategory(newCategory: ExpenseCategory) {
    selectedExpenseCategory = newCategory
}

fun validateInput(): Boolean {
    if (expenseTitle.isEmpty()) {
        errorMessage = "Title cannot be empty"
        return false
    }
    if (expenseAmount.isEmpty()) {
        errorMessage = "Amount cannot be empty"
        return false
    }
    if (expenseAmount.toDoubleOrNull() == null){
        errorMessage = "Amount must be a number"
        return false
    }
    if (errorMessage.isNotEmpty()) {
        return false
    }
    return true
}

fun loadExpenses() {
    expenses = dbHelper.getAllExpenses()
}

fun saveExpense(){
    errorMessage = ""
    if (validateInput()) {
        // Edit existing expense
        if (editingId != null) {
            dbHelper.updateExpense(
                id = editingId!!,
                title = expenseTitle,
                amount = expenseAmount.toDouble(),
                category = selectedExpenseCategory
            )
            editingId = null
        } else {
            dbHelper.insertExpense(
                title = expenseTitle,
                amount = expenseAmount.toDouble(),
                category = selectedExpenseCategory
            )
        }
        loadExpenses()
        clearExpense()
        showError = false
    } else {
        showError = true
    }
}



fun selectEditExpense(expense: Expense){
    editingId = expense.id
    expenseTitle = expense.title
    expenseAmount = expense.amount.toString()
    selectedExpenseCategory = expense.category
}

fun deleteExpense(expense: Expense) {
    dbHelper.deleteExpense(expense.id)
    loadExpenses()
}

fun clearExpense() {
    expenseTitle = ""
    expenseAmount = ""
```

```
        selectedExpenseCategory = ExpenseCategory.Food
        errorMessage = ""
    }

    fun seedData() {
        dbHelper.seedData()
        loadExpenses()
    }
}
```

ExpenseTrackerScreen.kt:

```
package com.example.expensetracker8959999.features.expensetracker

import androidx.compose.foundation.BorderStroke
import androidx.compose.foundation.border
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Row
import androidx.compose.foundation.layout.Spacer
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.height
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.text.KeyboardOptions
import androidx.compose.foundation.verticalScroll
import androidx.compose.material3.Button
import androidx.compose.material3.ButtonDefaults
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.OutlinedTextField
import androidx.compose.material3.RadioButton
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.input.KeyboardType
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.lifecycle.viewmodel.compose.viewModel
import com.example.expensetracker8959999.R
import com.example.expensetracker8959999.components.ExpenseForm
import com.example.expensetracker8959999.components.ExpenseList
import com.example.expensetracker8959999.data.ExpenseCategory
import com.example.expensetracker8959999.ui.theme.ExpenseTracker8959999Theme

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun ExpenseTrackerScreen(modifier: Modifier = Modifier) {
    val viewModel: ExpenseTrackerViewModel = viewModel()

    Column(
        modifier = modifier
            .padding(16.dp)
            .fillMaxWidth()
    ) {
        // Expense Form
        ExpenseForm(viewModel = viewModel)

        // Display error message if any
        if (viewModel.showError) {
            Text(
                text = viewModel.errorMessage,
                color = MaterialTheme.colorScheme.error,
                modifier = Modifier.padding(top = 8.dp),
            )
        }

        Spacer(modifier = Modifier.height(24.dp))

        // Expense List
        Text(
            text = stringResource(R.string.expense_list_title),
            style = MaterialTheme.typography.titleMedium,
            fontWeight = FontWeight.Bold
        )
        Spacer(modifier = Modifier.height(8.dp))
        ExpenseList(
            expenses = viewModel.expenses,
            onEdit = { viewModel.selectEditExpense(expense = it) },
            onDelete = {  viewModel.deleteExpense(expense = it) }
        )
    }


}

@Preview(showBackground = true)
@Composable
fun ExpenseTrackerScreenPreview() {
    ExpenseTracker8959999Theme {
        ExpenseTrackerScreen()
    }
}
```

MainActivity.kt:

```
package com.example.expensetracker8959999

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Scaffold
import androidx.compose.material3.Text
import androidx.compose.material3.TopAppBar
import androidx.compose.material3.TopAppBarDefaults
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.tooling.preview.Preview
import com.example.expensetracker8959999.features.expensetracker.ExpenseTrackerScreen
import com.example.expensetracker8959999.ui.theme.ExpenseTracker8959999Theme

class MainActivity : ComponentActivity() {
   @OptIn(ExperimentalMaterial3Api::class)
   override fun onCreate(savedInstanceState: Bundle?) {
      super.onCreate(savedInstanceState)
      enableEdgeToEdge()
      setContent {
         ExpenseTracker8959999Theme {
            Scaffold(
               modifier = Modifier.fillMaxSize(),
               topBar = {
                  TopAppBar(
                     title = {
                        Text(
                           text = stringResource(R.string.topbar_title),
                           modifier = Modifier.fillMaxWidth(),
                           style = MaterialTheme.typography.headlineLarge,
                           textAlign = TextAlign.Center,
                           fontWeight = FontWeight.Bold
                        )
                           },
                     colors = TopAppBarDefaults.topAppBarColors(
                        containerColor = Color(0xFF132543), // Dark blue
                        titleContentColor = Color.White
                     )
                  )
               }
            ) { innerPadding ->
               ExpenseTrackerScreen(modifier = Modifier.padding(innerPadding))
            }
         }
      }
   }
}
```

strings.xml

```
<resources>
   <string name="app_name">ExpenseTracker8959999</string>
```

```
    <string name="topbar_title">Expense Tracker</string>
    <string name="txtField_title">Expense Title</string>
    <string name="txtField_amount">Amount</string>
    <string name="trial">Test</string>
    <string name="txtField_category">Category</string>
    <string name="save_button">Save Expense</string>
    <string name="expense_list_title">Expense History</string>
</resources>
```